

(vim) 工作模式

手稿



封面

vim 有 2 种工作模式：命令模式和编辑模式，它们之间可以互相切换。

概述

vim 有以下 2 种工作方式：

- ◆ 命令模式（也称正常模式）

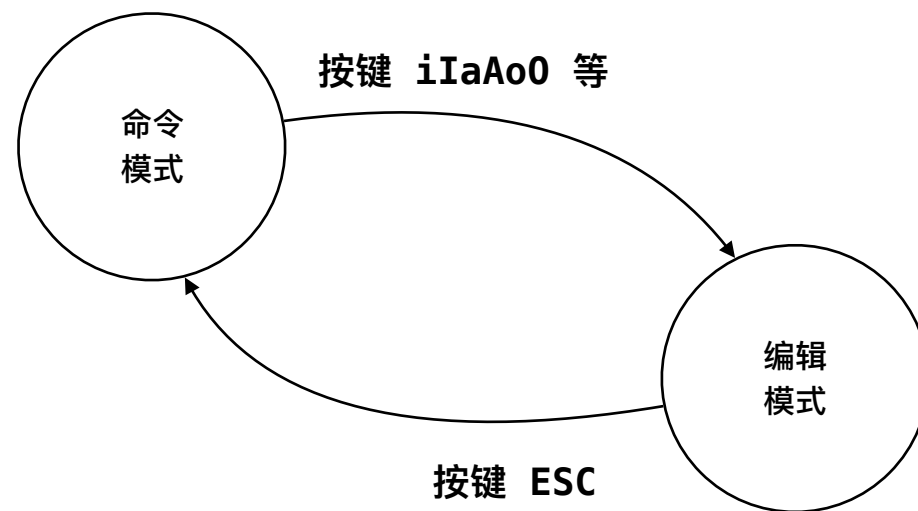
在命令模式下，vim 接受用户输入的命令（比如 :help）并作出回应。

- ◆ 编辑模式（也称输入模式）

在编辑模式下，vim 接受用户输入并显示在窗口，也就是说写入了缓冲区。

工作模式切换

vim 可以在命令模式和编辑模式之间进行切换，如下图所示。



命令模式

当 vim 会话开启后，默认进入命令模式。这时可以输入你想要执行的命令，它们有些不带前缀，例如 `a`、`yy`、`/` 等，而有些则需要加上冒号 (`:`) 前缀，例如 `:help`、`:w`、`:set` 等。

在输入带有前缀的命令时，vim 会在收到冒号后在窗口最下方一栏中显示输入的命令。如果输入过程中出现了错误，你可以用退格键删除然后重新输入，确认无误后按下回车执行命令。举例来说，为了让 C 语言代码更加美观或符合个人习惯，可以尝试用 `set` 命令做如下设置：

<code>:set number</code>	-- 显示行号
<code>:set autoindent</code>	-- 新行开始时自动缩进
<code>:set cindent</code>	-- 开启 C 程序自动缩进
<code>:set tabstop=4</code>	-- tab 使用的空格数
<code>:set shiftwidth=4</code>	-- cindent 中每步缩进使用的空格数
<code>:set showmatch</code>	-- 显示配对括号
<code>:set ruler</code>	-- 显示光标位置的行号和列号，用逗号分隔
<code>:set hlsearch</code>	-- 高亮显示匹配项

注：如果想要恢复某个选项的默认设置，可以在选项后面加上 `&`（比如 `:set ruler&`）。

```
45  **/  
46  :set autoindent
```

缩写形式

通常，命令和选项都有缩写形式。例如，`set` 可以缩写为 `se`，选项 `number` 可缩写为 `nu`。因此，上述设置可以简写为如下形式：

<code>:se nu</code>	-- 显示行号
<code>:se ai</code>	-- 新行开始时自动缩进
<code>:se cin</code>	-- 开启 C 程序自动缩进
<code>:se ts=4</code>	-- <code>tab</code> 使用的空格数
<code>:se sw=4</code>	-- <code>cindent</code> 中每步缩进使用的空格数
<code>:se sm</code>	-- 显示配对括号
<code>:se ru</code>	-- 显示光标位置的行号和列号，用逗号分隔
<code>:se hls</code>	-- 高亮显示匹配项

```
int main(int argc, char *argv[])
{
    do_something();
    printf("done\n");
    return 0;
}
:se nu
```

no 前缀

如果选项后面不带参数（比如 `ts` 后面的 `4`），你可以在选项前面加上 `no` 前缀获得相反的设置。例如：

<code>:se nonu</code>	-- 不显示行号
<code>:se noai</code>	-- 新行开始时不自动缩进
<code>:se nocin</code>	-- 不开启 C 程序自动缩进
<code>:se nosm</code>	-- 不显示配对括号
<code>:se noru</code>	-- 不显示光标位置的行号和列号
<code>:se nohls</code>	-- 不高亮显示匹配项

编辑模式

当 vim 切换至编辑模式后，窗口最下方一栏中显示 “-- INSERT --”，你可以在窗口中输入你想要的内容。例如，如果要在 `sailor.c` 的 `main` 函数 `return` 语句之前添加下面的语句表示工作结束：

```
printf("done\n");
```

你可以：

在命令模式下输入 `G` 来到文件尾，按下 2 次 `k` 键来到 `do_something` 一行，然后按下 `o` 键在下面添加一行，最后输入上面的 `printf` 语句；或者

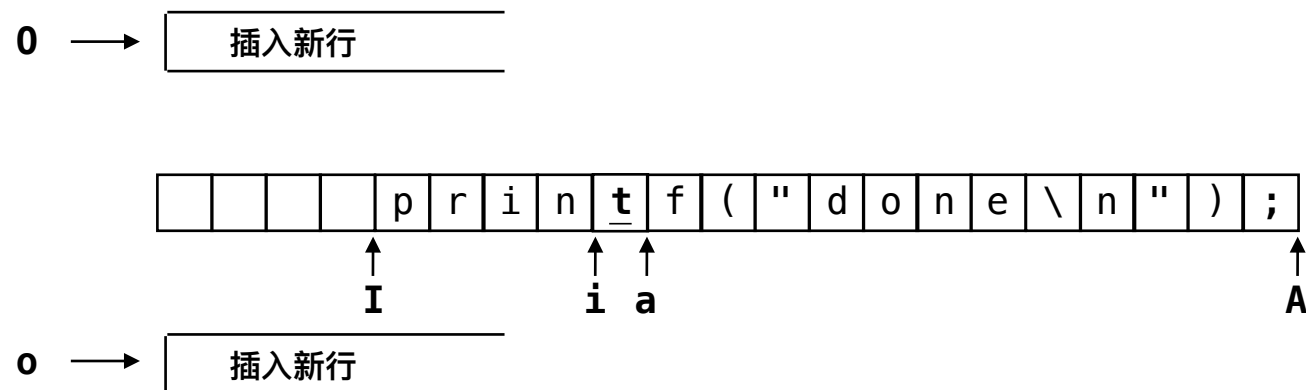
在命令模式下输入 `G` 来到文件尾，按下 `k` 键来到 `return` 一行，然后按下 `O` 键在上面添加一行，最后输入上面的 `printf` 语句；或者

在 `return` 一行开始处按下 `i/I/a` 其中一个按键，进入编辑模式后输入 `printf` 语句，最后回车。

```
64 int main(int argc, char *argv[])
65 {
66     do_something();
67     printf(
68     return 0;
69 }
-- INSERT --
```

进入编辑模式

下面是一些常用的进入编辑模式的命令，它们会在不同位置添加输入的内容。以前面新添加的 `printf` 一行为例，假设当前光标位于字符 `t` 处，那么按下：



- i -- 在 `t` 前面插入
- I -- 在行首（空白除外）插入，即 `p` 前面
- a -- 在 `t` 后面插入
- A -- 在行尾插入，即 `;` 后面
- o -- 在下一行插入（自动缩进）
- O -- 在上一行插入（自动缩进）

退出 vim

当结束编辑后，按下 `ESC` 返回到命令模式。

在命令模式下，你可以用不同命令选择保存当前修改、或者放弃当前修改，它们是：

<code>:wq</code>	-- 保存修改，退出 <code>vim</code>
<code>:q!</code>	-- 放弃当前修改，直接退出 <code>vim</code>

由于 `vim` 所有工作都在缓冲区内进行，因此保存修改意味着将缓冲区中的内容写入磁盘文件。